Department of Electrical & Computer Engineering

ECE 344L Microprocessors

Computer Types & the MIPS Microcontroller

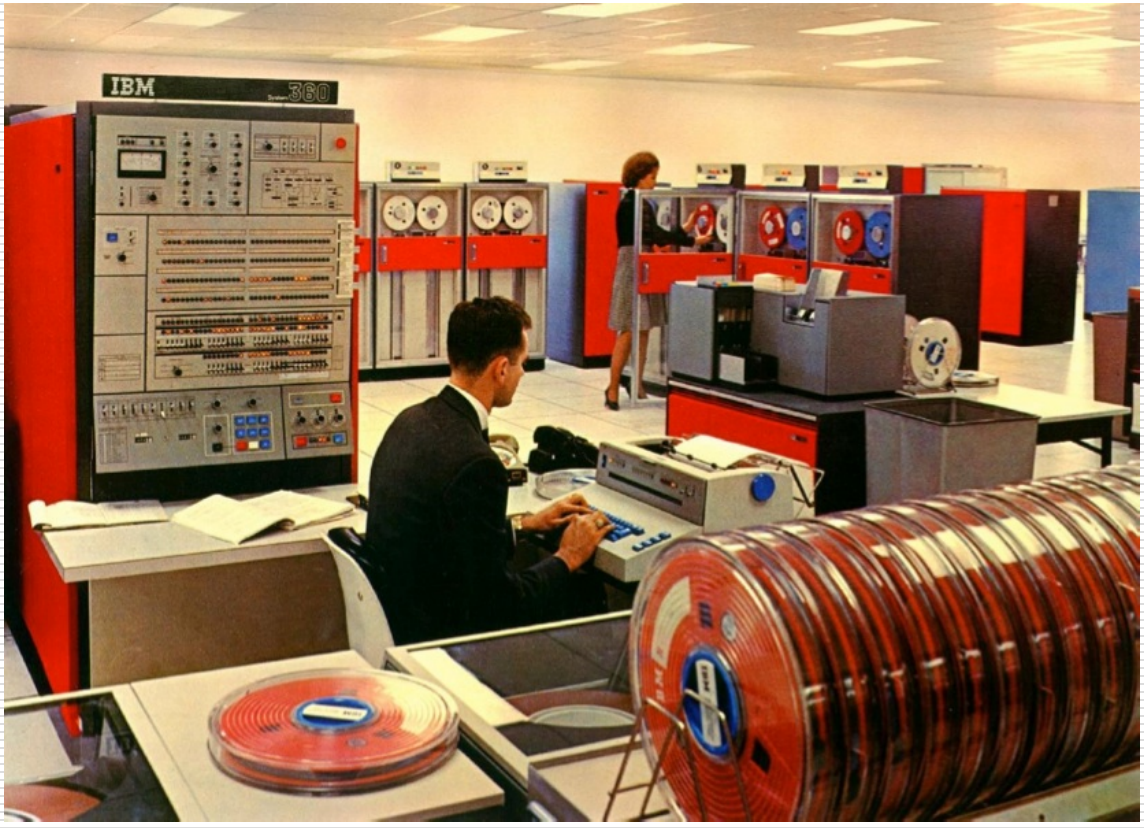
By

Dr. Edward Nava

Computer Configurations

- Mainframe
- Minicomputer
- Microprocessor
- Microcontroller
- Embedded Systems

Mainframe Computer



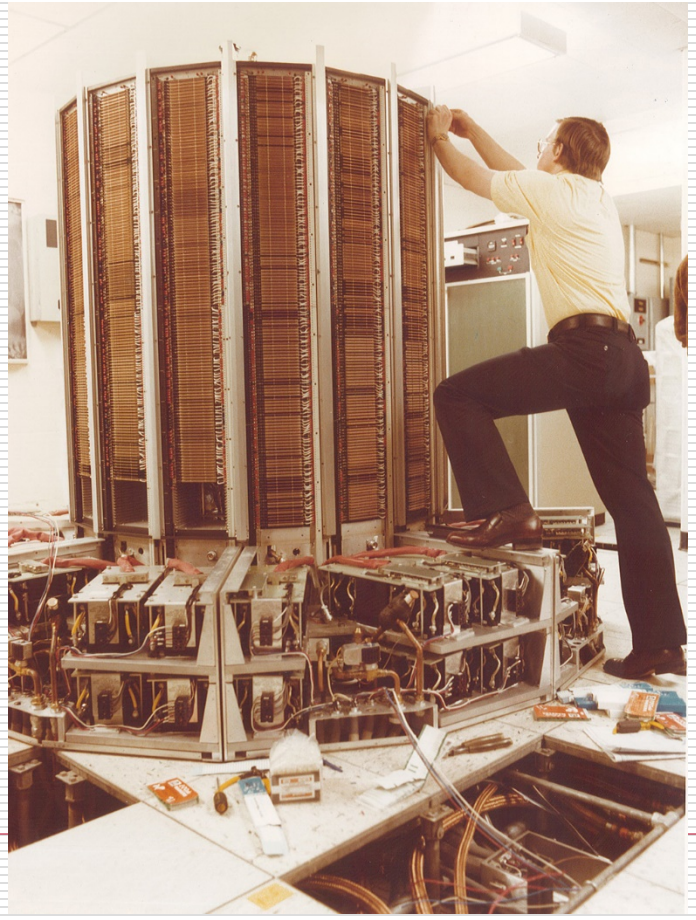
DEC Minicomputer

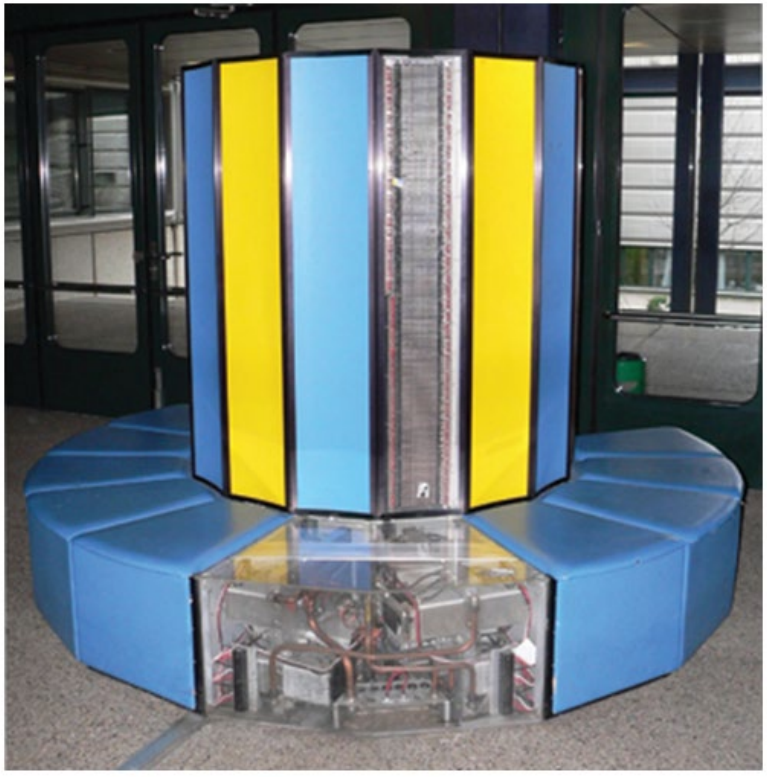
Data General Minicomputer

Personal Computer

Sun Workstation

Vintage Supercomputers





The first Cray computer ever delivered –
to Los Alamos National Laboratory in 1976.

Supercomputers

Principle Components

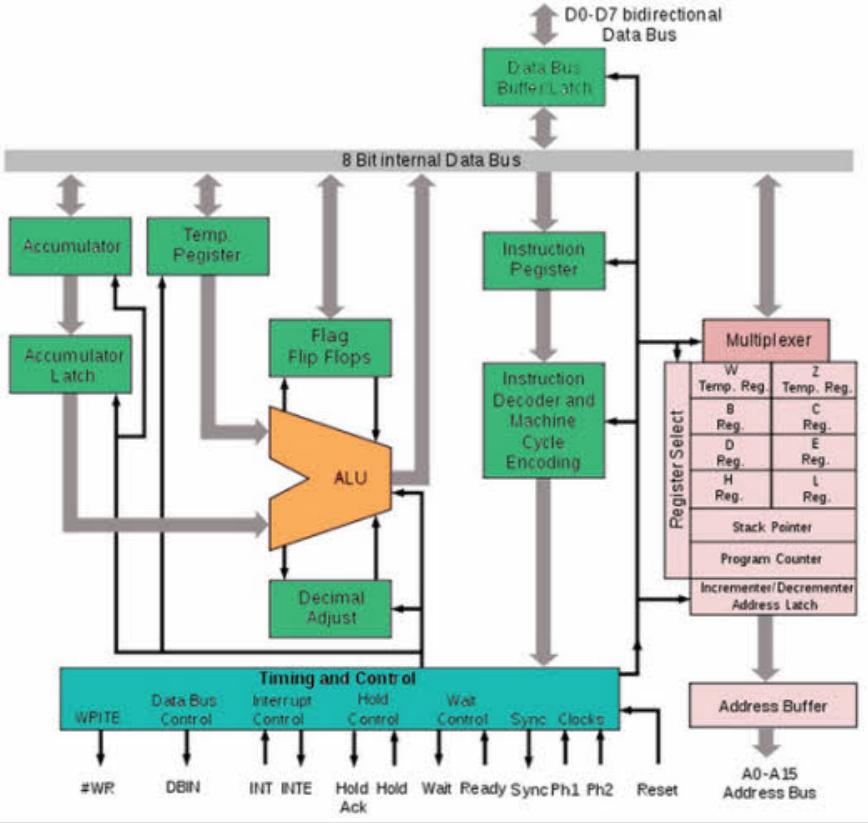
- Memory (Data, Instructions, …)
- Computation Engine
- Registers
- Control System
- I/O System
- Communication (wires, buses)

Modern Microprocessors

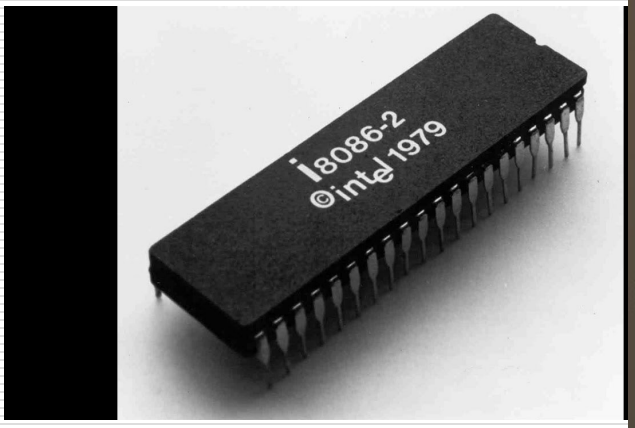
Integrated circuit version of a general purpose processor

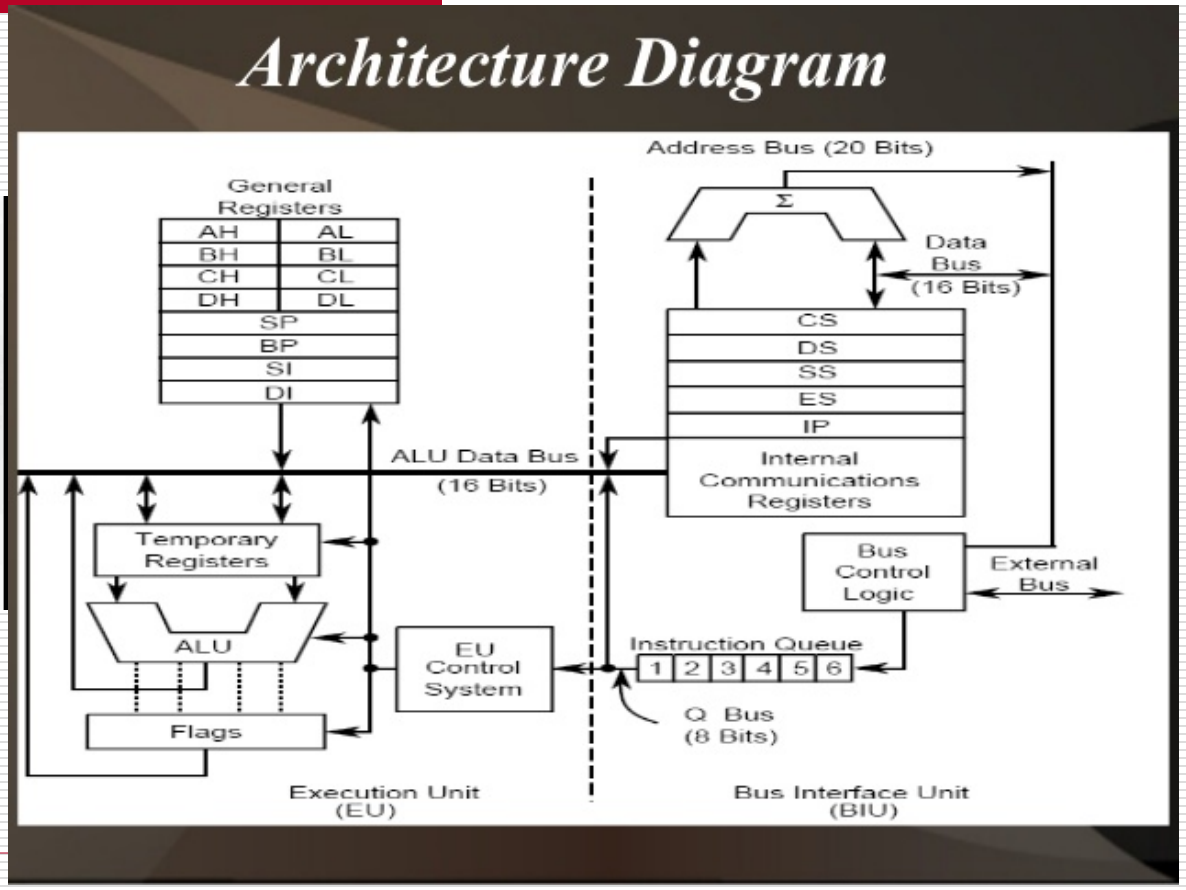
- Moderate-to-fast execution speed
- ALU/FPALU processing on chip
- Memory systems off chip
- Peripherals off chip and system dependent
- On-board caches for faster operation

8080 Microprocessor (1974 Vintage)

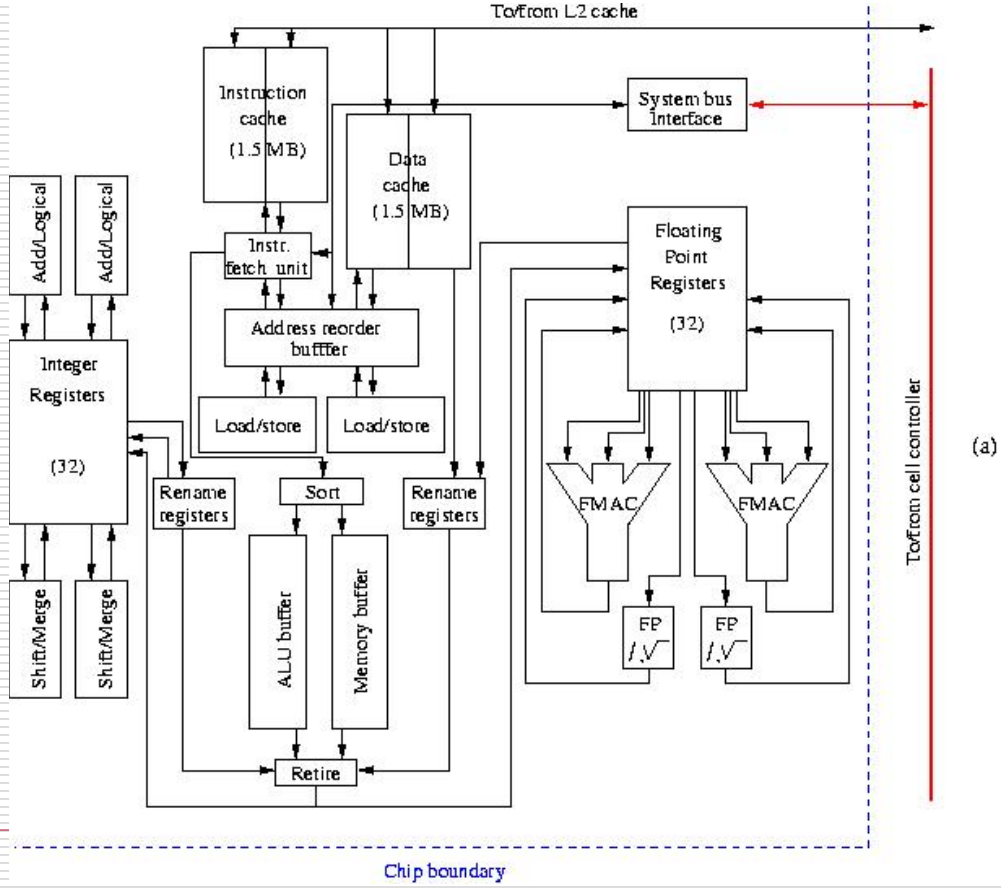


8086 Microprocessor (1979 Vintage)

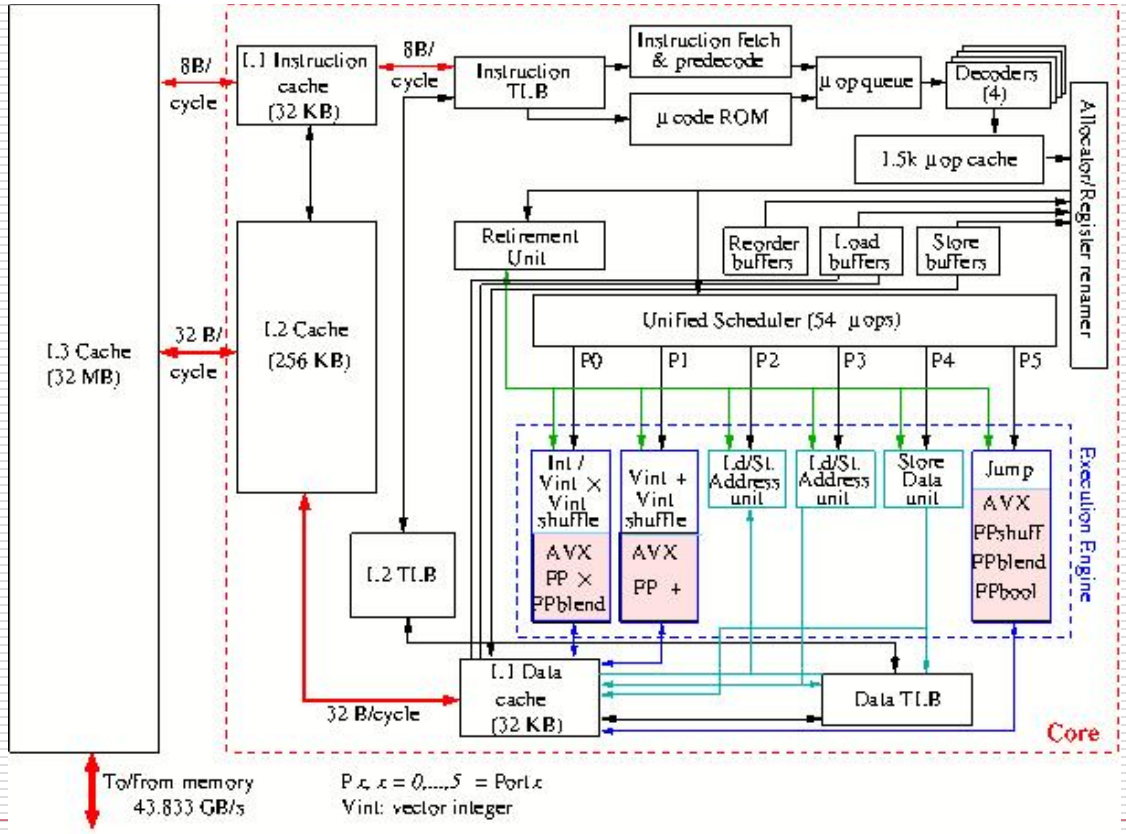




IA-32 Microprocessor (2001 Vintage)



IA-32 Microprocessor (2001 Vintage)



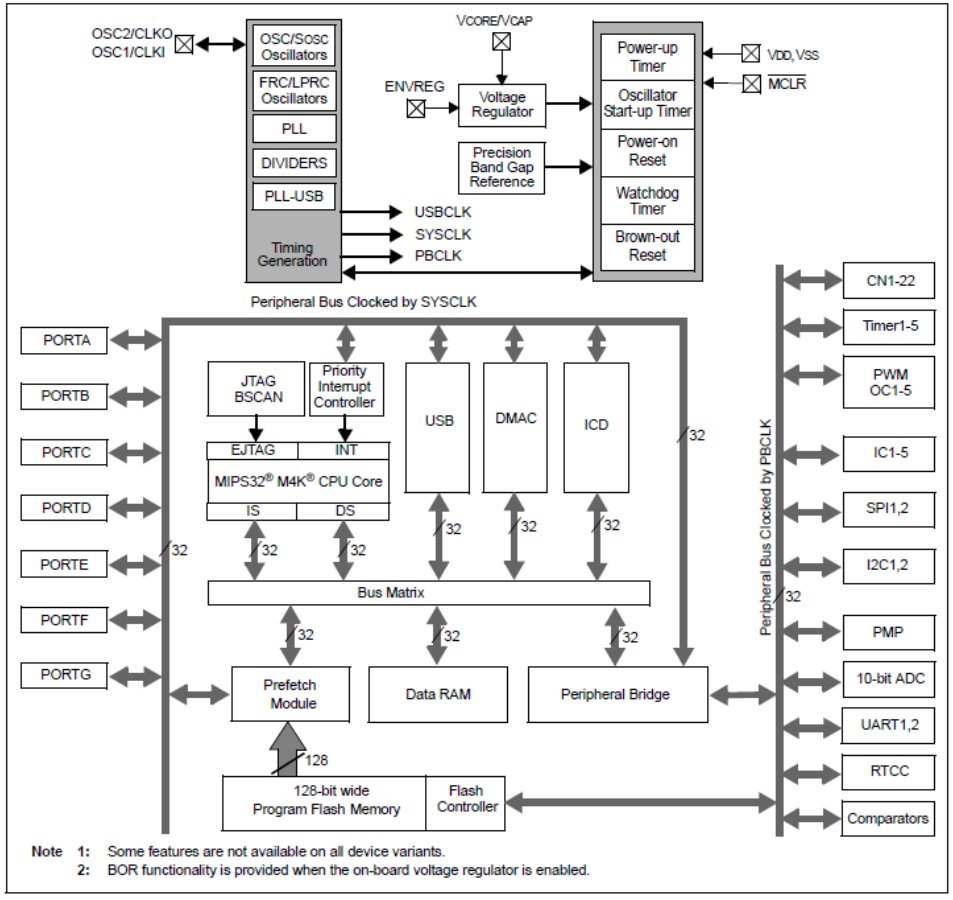
Microprocessor Heat Management System

Microcontroller

- Microprocessor with peripherals
  - On chip memory
  - On chip peripherals (UART/PWM/SPI …)
  - Low power modes
  - Minimal IC systems

Typical PIC32 Microcontroller

FIGURE 1-1: BLOCK DIAGRAM(1,2)



Note 1: Some features are not available on all device variants.
2: BOR functionality is provided when the on-board voltage regulator is enabled.

Embedded Systems

- A Microprocessor or microcontroller with potentially a mix of other components
  - Found in control, communication, ….
  - Interface with user may be non-traditional
  - Operating system is special purpose RTOS

How Do We Do Work in a Processor?

- The instruction set defines what's possible
- Work is accomplished by repeating instructions on data until the desired result has been obtained

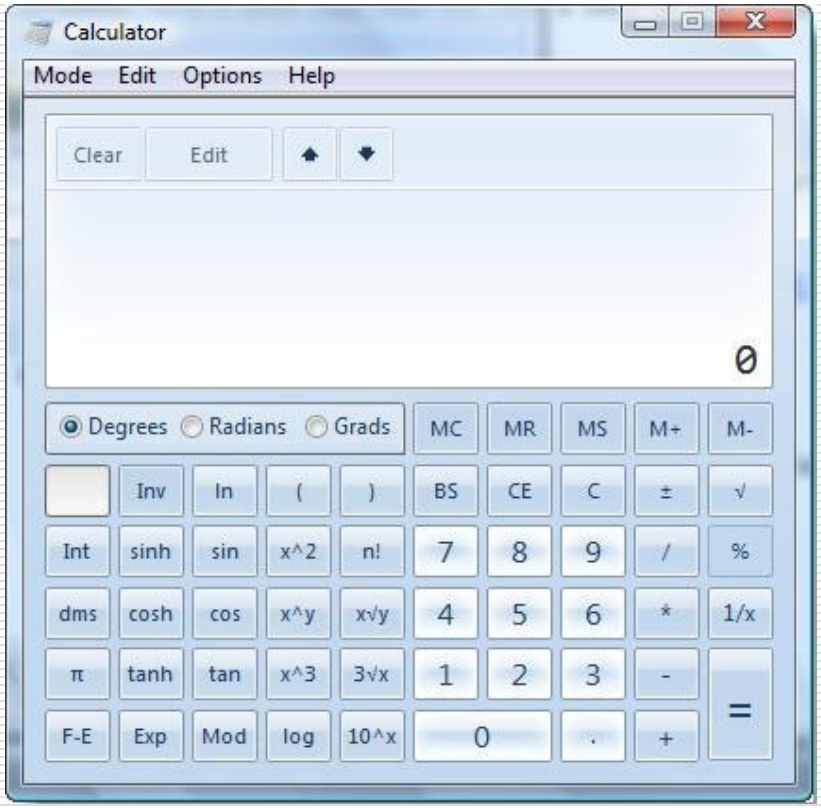
*The **Instruction Set Architecture** (ISA) is the part of the processor that is visible to the programmer or compiler writer.*

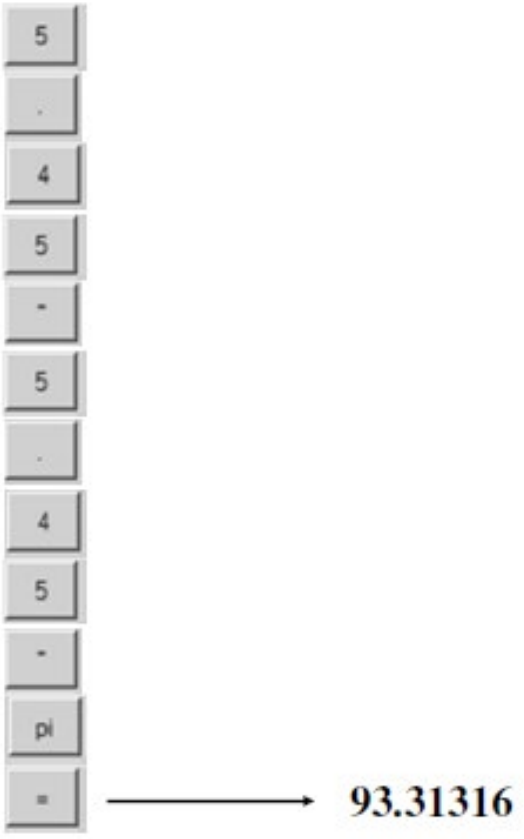
Calculator Example

The objective is to calculate the area of a circle whose radius = 5.45

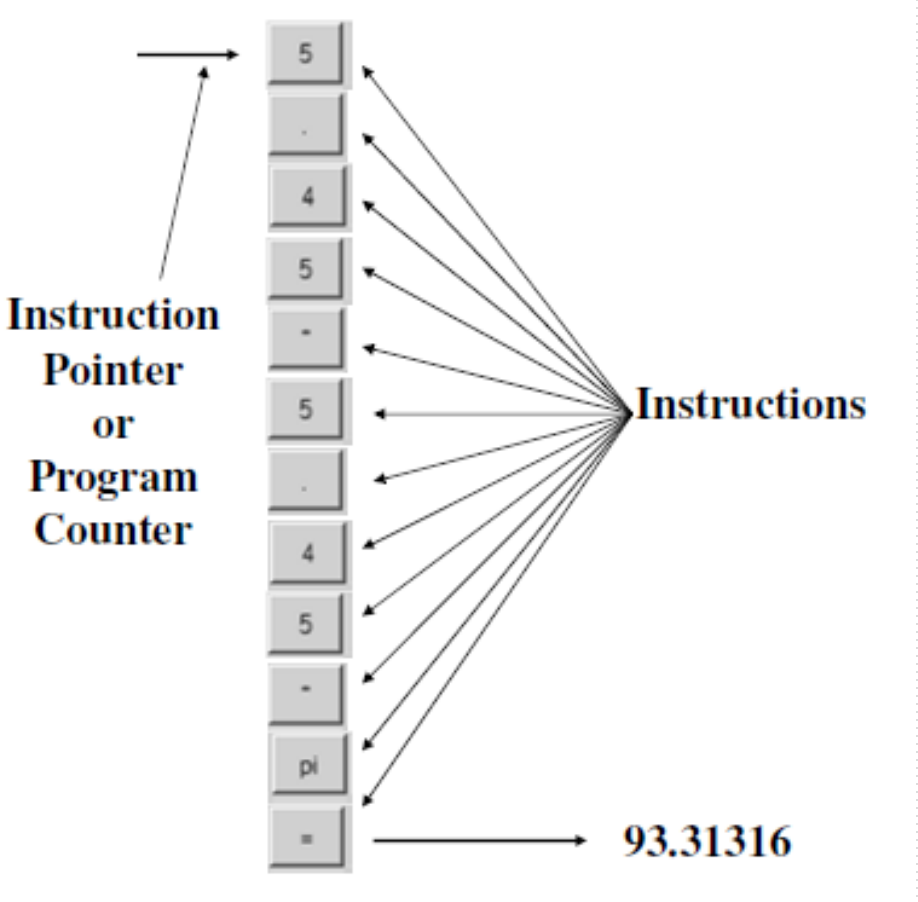
$$5.45 \times 5.45 \times \pi \quad = \quad 93.31316$$

Calculator Example –Instructions that will be used

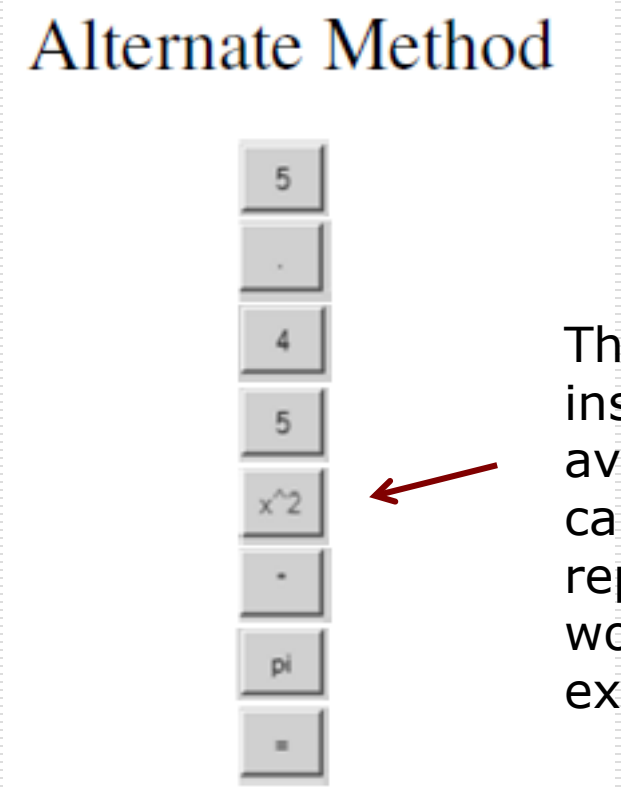




Calculator Example – Instruction Execution



Calculator – Revised Instruction Set



This "advanced" instruction may not be available on a basic calculator, which could represent a RISC ISA, but would be found in a more extensive CISC ISA

Computer Architecture

- The *instruction set* identifies the work (movement, transformation) that can be done in a particular architecture
- Any work to be done in a machine must be accomplished by a series of individual instructions
- The complexity of the instruction set determines how much work can be done by a single instruction ***Recall: CISC vs RISC***

Computer Architecture

- Work is done in computers by moving information from one place to another, sometimes with a transformation
  - Register to register
  - Register to memory
  - Memory to register
  - Memory to memory

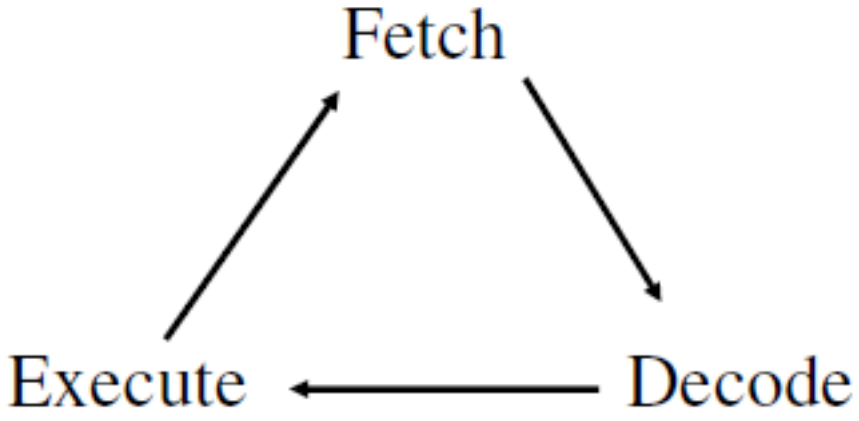
Common Registers in Processors

- Program counter (PC) – where is the program is executing
- Memory Address Register (MAR) – location of interest in memory
- Accumulator (ACC) – for a single address machine
- General purpose registers (Rn) – for data and addresses

More Common Registers in Processors

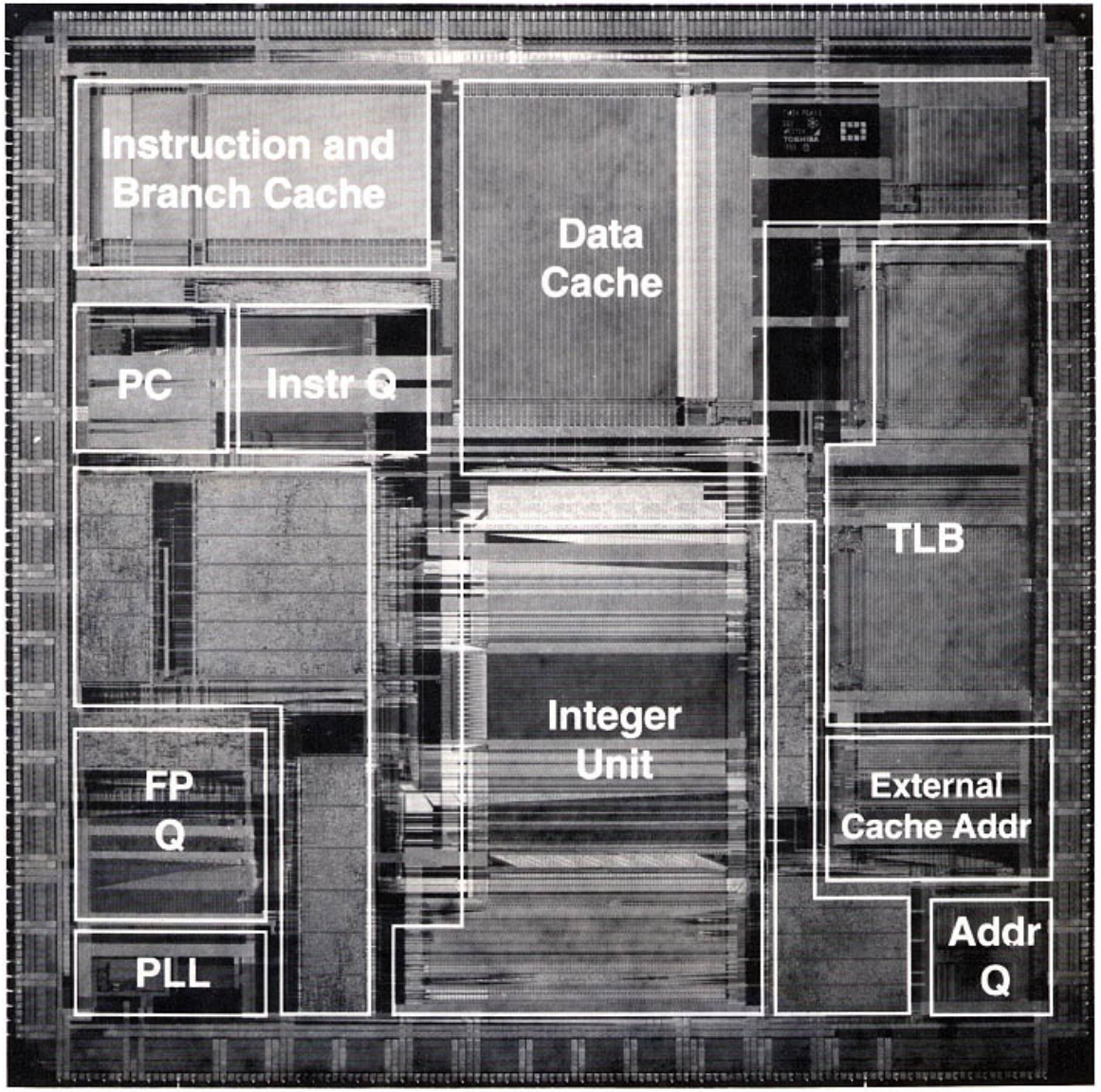
- Address registers (An) – point to location in memory
- Instruction Register (IR) – holds the instruction to be executed
- Status Register (SR) – holds information about the status of system
- System Control Registers – hold information about overall system operation

Basic operation in all computers



MIPS

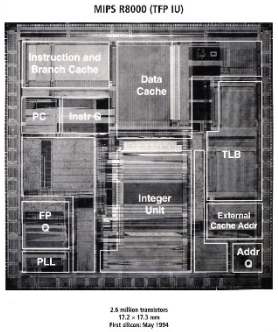
MIPS R8000 (TFP IU)



2.6 million transistors
17.2 × 17.3 mm
First silicon: May 1994

Why is the MIPS processor of interest to us?

- Going to the web site http://www.mips.com/ you will find that the MIPS processor is used in:
  - Cisco Routers
  - Laser Printers built by HP and Fuji Xerox
  - PDA's
  - Set-Top Boxes
  - Sony AIBO™ Entertainment Robot
  - Minolta Digital Camera
  - Sony PlayStation



MIPS Data Path Diagram

Program Counter (PC)

Memory

Out

Address

Data In

Instruction Register

Control Logic

Rs

Rt

Rd

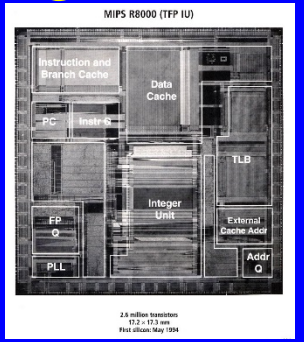
4

ALU

Register File

Fetch Execute Cycle

A Register Transfer Description of the Control Logic



IR = Mem[PC]

PC = PC + 4

Decode Instruction

Read from Reg. File

lw or sw

beqz

Address = Rs + Offset

R-Type

If (Rs == 0) then
PC = PC + Offset

sw

Memory[Address] = Rt

lw

Reg. File[Rt] = Memory[Address]

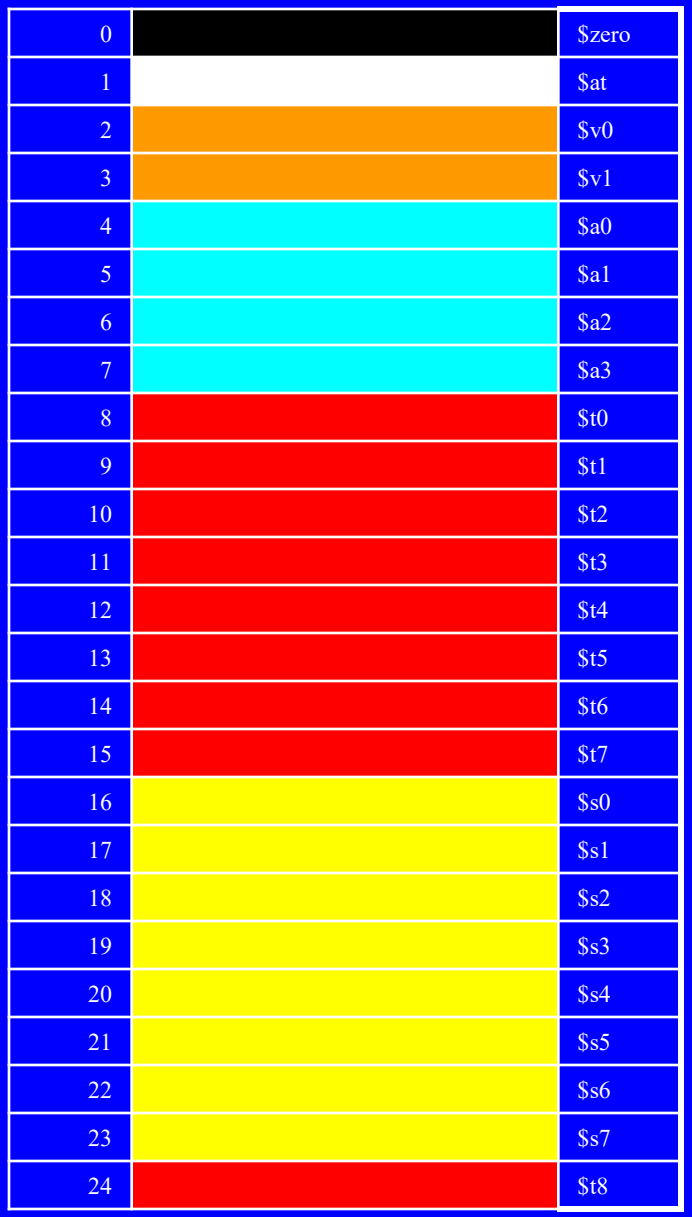
Reg. File[Rd] = Rs operation Rt

Registers in MIPS

- Rn – General Purpose registers (GPR)
  - 32 bit values
  - 32 registers total
  - Special assignment of function
  - R0 = 0
- Control Registers
  - 32 bit values
  - Variety of functions (SFR)

Register File

Register File



Return values from functions

Pass parameters to functions

Caller Saved Registers – Use these registers in functions

Callee-Saved Registers – Use these registers for values that must be maintained across function calls.

Number	Value	Name
25		$t9
26		$k0
27		$k1
28		$gp
29		$sp
30		$fp
31		$ra

System Memory

Used to hold Instructions, Data, and Operating System info

- Organized by byte (8 bits), half word (16 bits), or word (32 bits)
- Instructions can move bytes, half words, or words to or from memory
- Registers identify location in memory (either PC or Rn)
- MIPS enforces word address alignment

Memory Organization - Bytes

	00	01	02	03	04	05	06	07
0000	00	01	02	03	04	05	06	07
0008	08	09	0A	0B	0C	0D	0E	0F
0010	10	11	12	13	14	15	16	17
0018	18	19	1A	1B	1C	1D	1E	1F

Memory Organization – Half words

	00	02	04	06
0000	0001	0203	0405	0607
0008	0809	0A0B	0C0D	0E0F
0010	1011	1213	1415	1617
0018	1819	1A1B	1C1D	1E1F

Memory Organization - Words

	00	04
0000	00010203	04050607
0008	08090A0B	0C0D0E0F
0010	10111213	14151617
0018	18191A1B	1C1D1E1F

MIPS Instruction Formats

The MIPS instruction set has three basic formats:

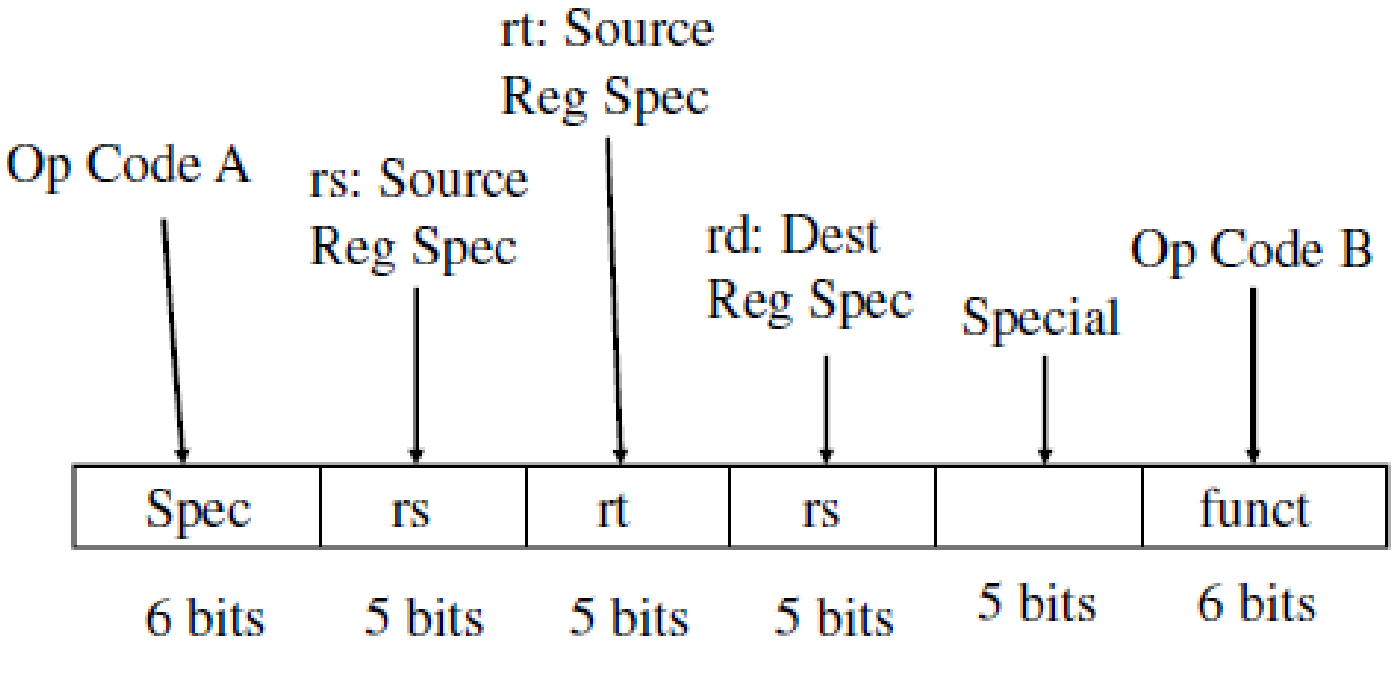
- Register
- Immediate
- Jump instructions

Register Format

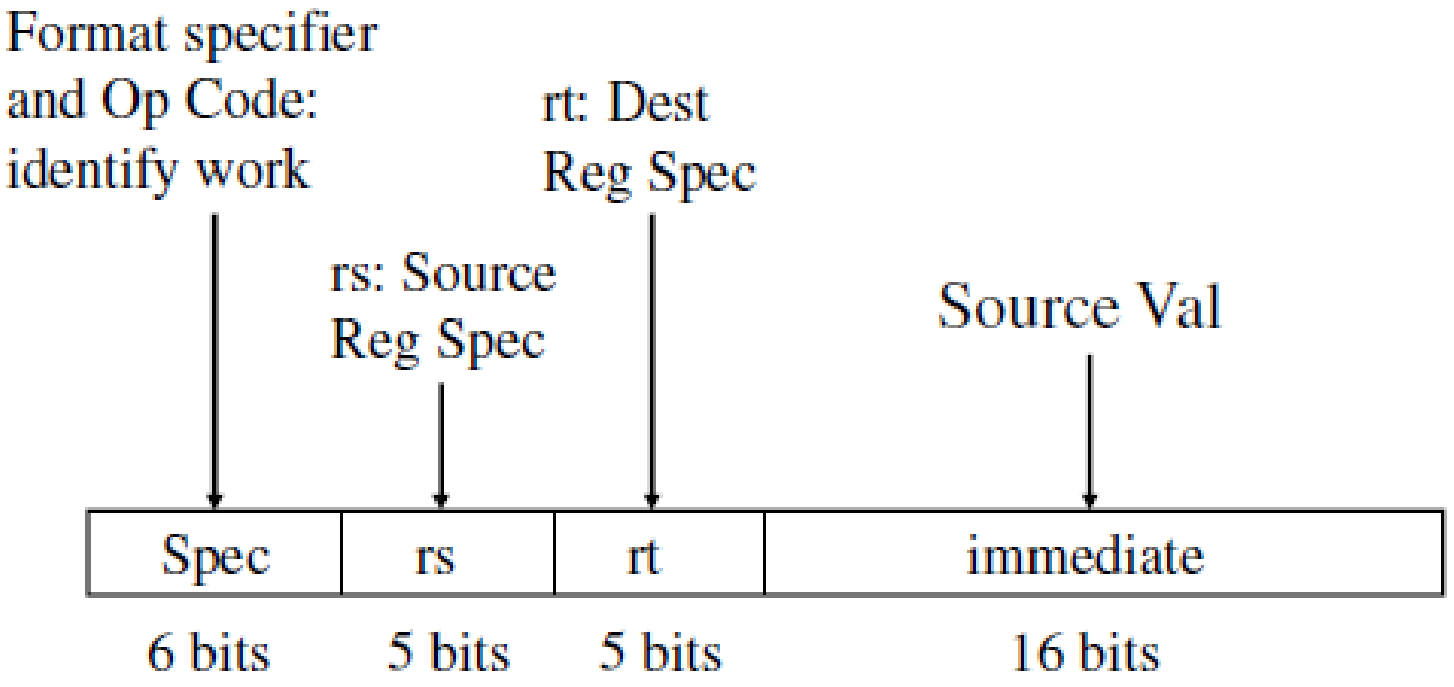
Used for work instructions which perform operations using data registers

- 3 address format for ADD, SUB, etc.
- 2 address format for some instructions
- some formats can have constant value for one of the arguments
- Instructions must provide bits to identify function (op code) and bits to identify target register(s)

Register Format



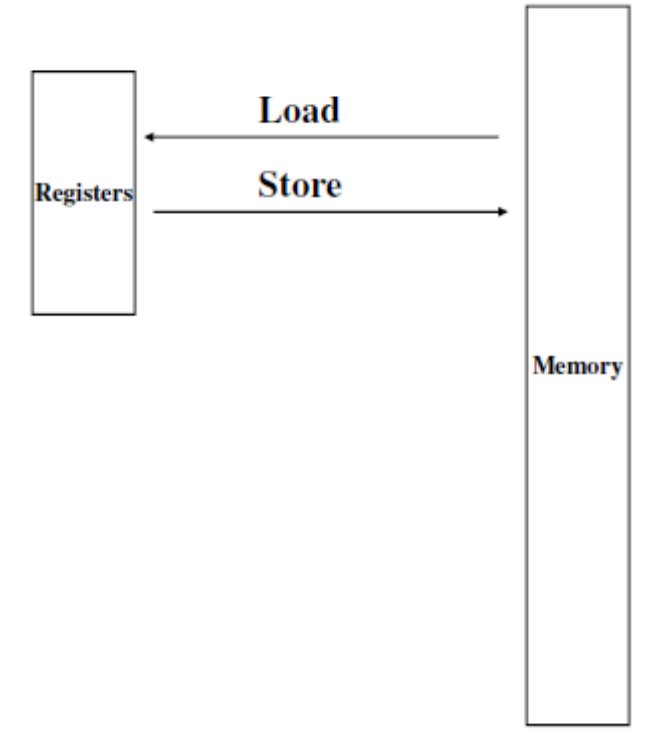
Immediate Format



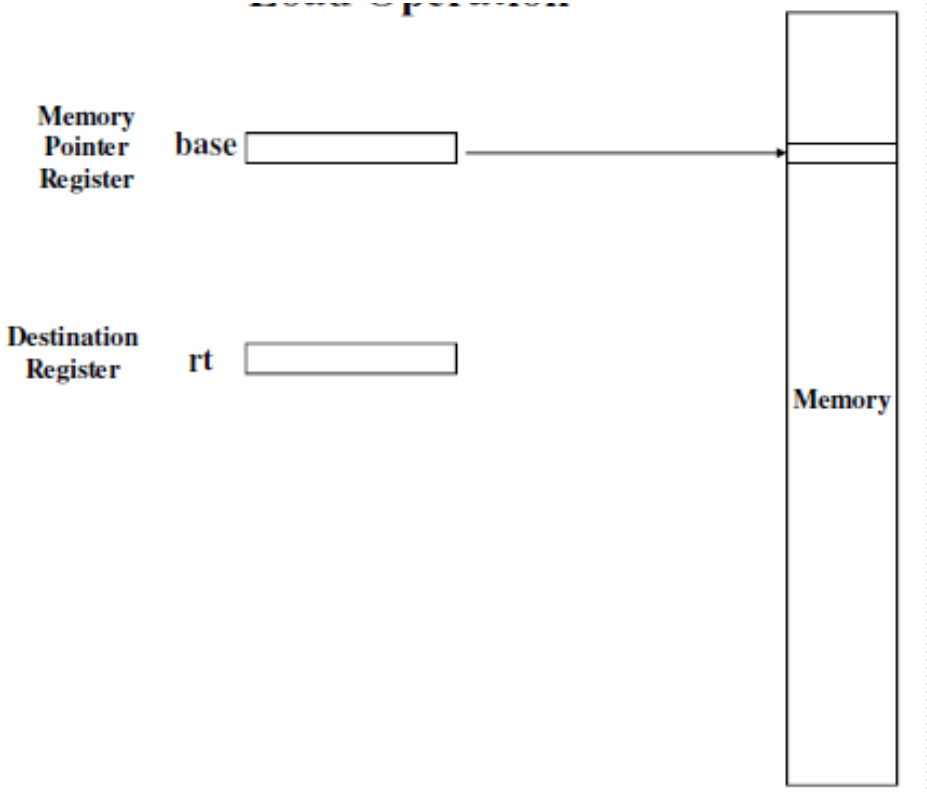
Movement Instructions

- Simple movement : register to register
- Load: memory to register
- Load: constant to register
- Store: register to memory
- Dual mode instructions: movement and work

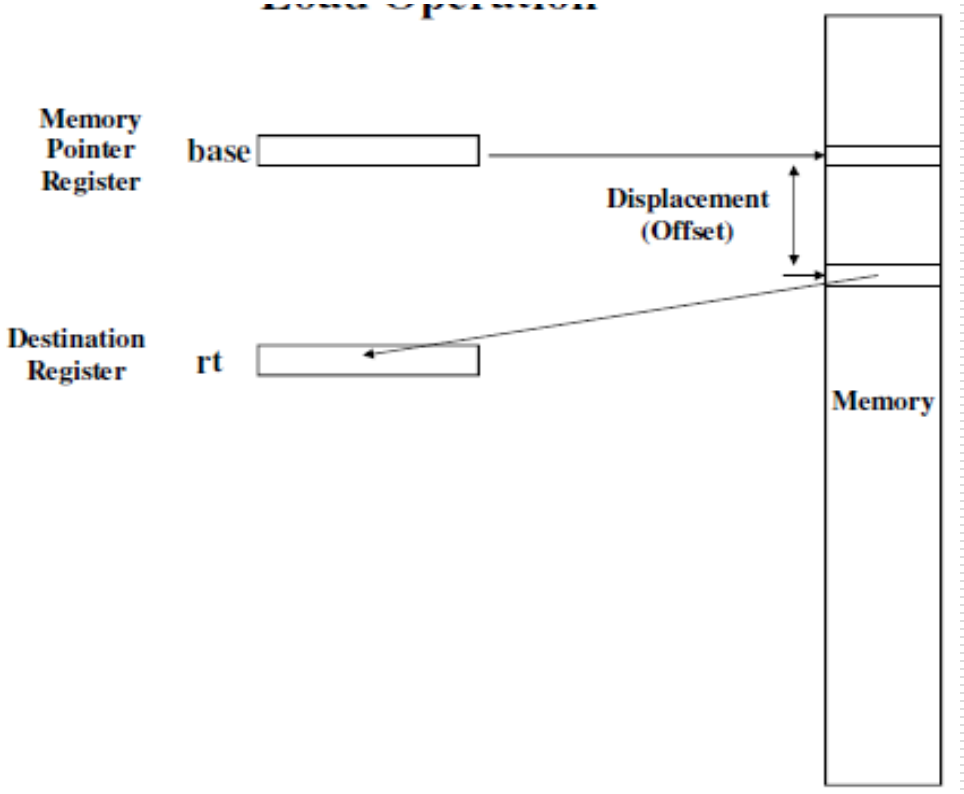
Data Movement



Data Movement - Addressing



Data Movement – Addressing (2)



Jump Instruction

opcode	target
00001f	t t t t t t t t t t t t t t t t t t t t t t t t t t